Using a 3-D Shape Model for Video Coding

Chuo-Ling Chang, Peter Eisert[1], and Bernd Girod

Information Systems Laboratory
Stanford University, CA 94305, U.S.A.
{chuoling,bgirod}@stanford.edu

Abstract

In this work, we present a model-aided coder that incorporates 3-D shape and motion information to improve the performance of a video coding system. The 3-D shape model of a rigid body is estimated from the video sequence directly. Using the shape model, the 3-D motion of the rigid body and the background global motion are also estimated for video coding. Experimental results show that bit-rate savings of 30%-40% are achieved at equal PSNR compared to the H.26L test model TML-8.0. This corresponds to 2-3 dB improvements in PSNR when encoding at the same bit-rate.

1 Introduction

Model-aided coding uses knowledge of the 3-D structure in the scene and the camera motion in order to encode video sequences more efficiently. This knowledge has been incorporated into a traditional hybrid waveform video coder by using a synthesized model frame as one of the reference frames for multi-frame motion-compensated prediction. It has been shown to significantly improve the coding efficiency for head-and-shoulder sequences in video-conferencing [1].

For more general classes of video sequences, research has also been conducted to encode the sequences by exploiting the 3-D information [2] [3][4]. In [2], views of a static scene are synthesized by view morphing, which implicitly makes use of the 3-D geometry of the scene. In [3], a segment of a video sequence with a static scene is considered. 3-D shape and motion in the segment is estimated and used to approximate the entire segment from the starting frame. In [4], sequences containing a collection of rigid bodies are considered. 3-D shape and motion of each rigid body is estimated, and each frame is synthesized from the reconstructed previous frame using the 3-D information. The residual in regions that the model does not predict well is additionally coded.

In this work, we consider video sequences containing one rigid object of primary interest. No assumption is made for other parts of the scene. The intrinsic camera parameters are assumed to be known and remain constant through the sequence. For video coding, the 3-D shape of the object is estimated and used in the model-aided coding framework. The 3-D motion of the object as well as the global motion in the background is estimated to facilitate motion-compensated prediction. Experimental results are compared with the H.26L test model TML-8.0 coder [5].

2 Model-Aided Coding

The overall structure of model-aided coding is shown in Figure 1.

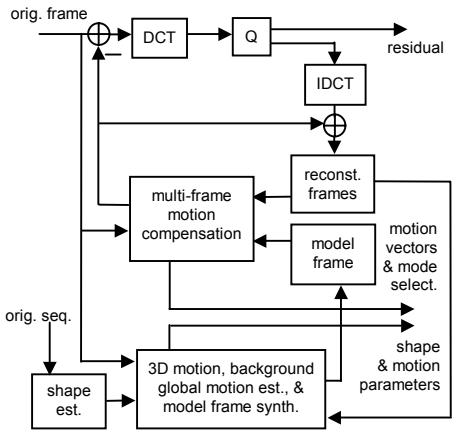

Figure 1. Model-aided coding structure

[1] now with Heinrich-Hertz-Institute, Berlin, Germany. Email: eisert@hhi.de

The 3-D shape S of a rigid body object is first estimated from the original video sequence. To encode frame F_t, 3-D motion of the object between F_t and F_{t-1}, denoted by $mo_{t,t-1}$, is estimated. This also yields a segmentation of the object from the background in F_t. The background is approximately modeled as a 3-D planar object and its global motion between F_t and F_{t-1}, denoted by $mb_{t,t-1}$, is estimated. With the shape model, motion information, and the texture from reconstructed previous frame F'_{t-1}, a model frame M_t can be synthesized to approximate F_t. This M_t serves as one of the reference frames for multi-frame motion compensated prediction along with reconstructed previous frames F'_{t-1}, F'_{t-2}…[6]. For decoding the sequence, *mo* and *mb* need to be transmitted along with the residual, motion vectors, and mode selection information for every P-frame. The shape model is transmitted only once in the beginning of the sequence. 3-D shape estimation, object 3-D motion estimation, background global motion estimation, model frame synthesis, and multi-frame motion-compensated prediction are described in the following sub-sections.

2.1 3-D Shape Estimation

In model-aided coding, the approximate 3-D shape of the object is represented by a triangle mesh model. The camera intrinsic parameters are first estimated using a calibration object with known geometry. Several frames equally spaced through the sequence, i.e., F_{nd}, where n is a positive integer and d denotes the spacing, are chosen as key-frames. By identifying salient feature points on the object and establishing correspondences for the key-frames, classical structure-from-motion algorithms can be applied to simultaneously estimate the 3-D positions of feature points as well as the object poses (or the relative camera poses) at the key-frames [7]. A triangle mesh of the shape model S is constructed from the feature points, and object poses p_{nd} at the key-frames can be used to initialize the process for estimating object motion at every frame as will be described in Section 2.2.

Note that in this work, we have selected the feature points and established correspondences by hand. Commercial software is used to estimate the 3-D shape and object poses from the correspondences given the estimated camera intrinsic parameters [8]. Several techniques that automatically reconstruct 3-D shape from uncalibrated camera views could instead be applied here [9][10]. Furthermore, for sequences containing buildings and industrial products designed by CAD tools, the shape model could be provided directly for model-aided coding.

2.2 Object 3-D Motion Estimation

Under the assumption of a rigid body with a Lambertian surface and given the estimated camera intrinsic parameters, shape model S and object pose p_{nd} at a key-frame F_{nd}, object 3-D motion between F_{nd+1} and F_{nd}, i.e., $mo_{nd+1,nd}$, can be estimated by a model-based 3-D motion estimation approach as used in [9]. Object pose p_{nd+1} can then be computed directly by accumulating $mo_{nd+1,nd}$ over p_{nd}. By applying this frame-by-frame, *mo*'s between every pair of adjacent frames and hence p's at every frame can be obtained.

The model-based approach is based on the optical flow equation:

$$\boldsymbol{G}_{t-1,k}{}^{T}\boldsymbol{d}_{t-1,k} = I_{t-1,k} - I_{t,k}, \forall k \qquad (1)$$

where $I_{t-1,k}$ and $I_{t,k}$ denote intensity of an object point k in frame F_{t-1} and F_t, $\boldsymbol{d}_{t-1,k}$ is the displacement vector of point k from F_{t-1} to F_t, and $\boldsymbol{G}_{t-1,k}$ is the average spatial gradient vector for point k over F_{t-1} and F_t.

$\boldsymbol{d}_{t-1,k}$ can be described as a function of $mo_{t,t-1}$ and can be linearized. Therefore, an over-determined linear system is established and the 6 parameters of $mo_{t,t-1}$ can be solved by linear regression. Note that the boundary of the object is excluded from estimation to improve stability.

Due to the inherent linearization in the optical flow equation and the linear approximation mentioned above, the solution is valid only for small displacements between the two frames. This problem is overcome by using an analysis-by-synthesis loop with multi-resolution structure [9]. Note that model frame synthesis, which will be discussed in Section 2.4, is also incorporated in the analysis-by-synthesis structure in the estimation stage.

Parts of the object surface that are specular violate the Lambertian surface assumption and should be treated as outliers. Outlier removal is done

using a soft-threshold technique. The linear system is first solved using linear regression as mentioned above. Because the linear system is over-determined, the solution does not satisfy (1) for all points. The deviation from the equality in (1) is used as an error measurement for each point. A new linear system is formed by giving smaller weights to the optical flow equations of the points with higher error measurement, and this system is solved again by linear regression. Hence, potential outliers have reduced effects on the overall system.

2.3 Background Global Motion Estimation

From the camera intrinsic parameters, shape model and object 3-D motion, the projection of the object onto the image plane can be computed and the object is segmented from the background. The background is modeled as a 3-D planar object, which is a good approximation for background in the distance or with little depth variation. A projective transformation:

$$x_{t-1} = \frac{a_0 x_t + a_1 y_t + a_2}{a_6 x_t + a_7 y_t + 1} \quad y_{t-1} = \frac{a_3 x_t + a_4 y_t + a_5}{a_6 x_t + a_7 y_t + 1} \qquad (2)$$

describes the projected 3-D motion of a planar object on the image from F_t to F_{t-1} without knowledge of the initial pose of the object. (x_{t-1}, y_{t-1}) is the position of a background point at F_{t-1}, while (x_t, y_t) is the position of the same point at F_t.

Transformation from F_t to F_{t-1}, i.e., $mb_{t,t-1}$, which consists of $a_0, \ldots, a_7$, is estimated. In this case, $\boldsymbol{d}_{t-1,k}$ in (1) is simply a function of $a_0, \ldots, a_7$. By rearranging the terms, an over-determined linear system with unknown $a_0, \ldots, a_7$ is established. Therefore, $mb_{t,t-1}$ can be again estimated in the same fashion as $mo_{t,t-1}$.

Note that both object 3-D motion and background projective transformation can be estimated either between original frames, i.e., F_{t-1} and F_t, or using one reconstructed frame, i.e., F'_{t-1} and F_t. For lower bit-rate, the estimation using the latter approach tends to be unreliable because of the large quality difference between the original frame and the reconstructed frame.

2.4 Model Frame Synthesis

The model frame M_t, that should approximate F_t, is synthesized as follows. Given the estimated shape S and object pose p_t, the 3-D position of an object point at (x_t, y_t) in M_t can be computed. This point is then moved in 3-D according to the object motion, and projected onto F'_{t-1} at (x_{t-1}, y_{t-1}). The intensity at (x_{t-1}, y_{t-1}) is interpolated from F'_{t-1} and used at (x_t, y_t) in M_t. The same procedure is applied for the background where (x_{t-1}, y_{t-1}) can be directly computed from (2).

Due to the approximate representation using triangle mesh, the projected shape model tends to be smaller in the image plane than the actual image of the object. Therefore, most pixels at the object boundary are not covered by the shape model. To overcome this problem, pixels outside but near the boundary of the shape model are considered as part of the object for model frame synthesis, their 3-D position is extrapolated from neighboring pixels covered by the shape model.

Note that in [1], a global texture map for the object is transmitted separately to render all model frames in the sequence, while the structure in Figure 1 uses the reconstructed previous frames to synthesize the model frame. The advantage of the latter approach is that the constant intensity property for Lambertian surface is better held between F'_{t-1} and F_t than between the global texture map and F_t. Therefore, M_t resembles F_t more, which is beneficial for both motion-compensated prediction and motion estimation with an analysis-by-synthesis structure. The drawback of the latter approach is error propagation for object motion estimation. Specifically, once the estimated object pose p_{t-1} is biased, the intensity value for M_t extracted from F'_{t-1} is also biased and hence the error propagates. Estimation and synthesis from multiple previous frames can alleviate this problem by suppressing the effects of biased frames, similar to outlier removal mentioned in Section 2.2.

2.5 Multi-Frame Motion-Compensated Prediction

The reconstructed previous frames and the model frame both serve as reference frames for motion-compensated prediction. The coder decides which reference frame should be used for each macroblock by minimizing a Lagrangian cost function

[1][6]. When the model frame fails to provide good prediction, the coder selects the prediction provided by the reconstructed previous frames. In other words, object 3-D motion and background projective transformation are used as additional motion models. The coder can select the most suitable motion model for each macroblock among the additional models, pure translation, and combination of the two. Therefore, it exploits the 3-D shape and motion information only when it gives better results than a traditional waveform coder.

3 Experimental Results

Experiments are conducted with two hand-held-recorded 352x240 YUV 4:2:0 sequences *Digital Camera* and *Car* at a frame rate of 15 fps with known constant focal length. First frame of each sequence is shown in Figure 2. Both sequences contain a rigid body in the foreground. In *Digital Camera*, the object is rotated and shifted. There is also camera motion. In *Car*, the camera moves around the object in addition to the motions of background objects (trees, cars passing by).

3.1 Shape and Motion Estimation

To estimate the object shape, frame 0, 100, 200, 300, 400, and 499 of *Digital Camera* are chosen as key-frames. By selecting feature points and establishing correspondences in the key-frames, 3-D positions of the feature points are estimated, as well as object poses at the key-frames. The feature points then serve as vertices of triangle meshes in the shape model. The same procedure is applied for *Car*, but only one side of the object is reconstructed and used. Resulting shape models are shown in Figure 2. Using the shape model, model-based 3-D motion estimation is applied to estimate object motion at every frame as described in Section 2.2. The projective transformation that describes the motion of the background is also estimated for every frame as discussed in Section 2.3.

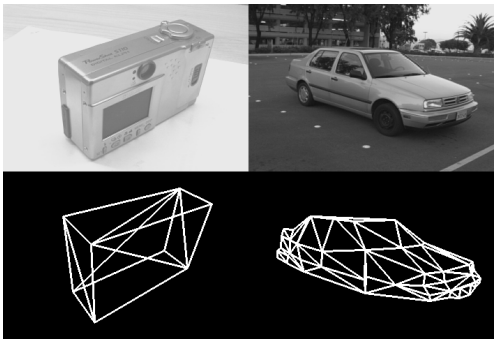

Figure 2. (Left) frame 0 and the estimated shape model for *Digital Camera* (Right) frame 0 and the estimated shape model for *Car*.

3.2 Video Coding

The estimated 3-D shape model, object motion, and background projective transformation are used to synthesize a model current frame from a reconstructed previous frame. The difference between the original current frame and either the reconstructed previous frame or the model current frame is shown in Figure 3. It can be seen that the model current frame closely approximates the original current frame.

The H.26L test model TML-8.0 coder has been modified to incorporate the model-aided coding (MAC) structure [5]. Two reference frames are

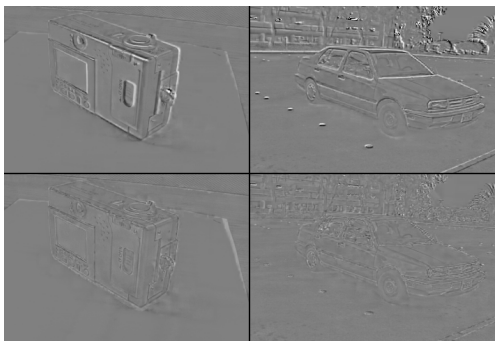

Figure 3. Difference images (gray denotes small differences, black/white denote large differences)
(Left) *Digital Camera*: Top shows differences between original frame 110 and reconstructed frame 109. Bottom shows differences between original frame 110 and model frame 110.
(Right) *Car*: Top shows differences between original frame 10 and reconstructed frame 9. Bottom shows differences between original frame 10 and model frame 10.

used for multi-frame motion-compensated prediction; the reconstructed previous frame is used as the first reference frame and the model frame is used as the second one. Block-based translational motion compensation is applied to both reference frames, and the final prediction frame is formed by minimizing the Lagrangian cost for each macroblock [1][6].

We also extended the TML-8.0 coder to include a *COPY* mode for both reference frames, rather than only for the first reference frame. The *COPY* mode directly copies a macroblock from one of the reference frames without coding any residual and motion vector [5]. One additional bit indicating the reference frame selection for the *COPY* mode is added in the modified coder.

Side information also needs to be encoded. In the experiments, object 3-D motion is represented by the quantities t_x, t_y, t_z, a_x, a_y, and θ, where (t_x, t_y, t_z) is the 3-D translation, (a_x, a_y) describes the 3-D rotation axis normalized to unit length, and θ is the rotation angle. One set of 3-D motion parameters is quantized and fixed-length encoded with 118 bits, including 18 bits for each of t_x, t_y, t_z, 21 bits for each of a_x, a_y, and 22 bits for θ. However, the bit-rate could be significantly reduced by coarser quantization, predictive coding, and entropy coding. To encode the projective transformation, four coordinates are pre-selected and are known both in the encoder and decoder. By applying the transformation to the four coordinates, four 2-D displacement vectors are obtained and can be used to represent the transformation. This parametrization is more robust to quantization than the 8 parameters in (2). It is quantized and encoded by 56 bits per set. A shape model is transmitted only once in the beginning of the sequence. Therefore, overhead for the shape model is not considered in the following discussion.

Rate-PSNR curves for the original TML-8.0 and for model-aided coding (MAC) are shown in Figure 4. 200 frames of *Digital Camera* and 150 frames of *Car* are encoded. Both TML-8.0 and MAC use 1/8 pel accuracy for block-based motion compensation. PSNR is computed from the luminance component only. Only one I-frame is used. Rate is expressed as bits per P-frame, and the bit-rate for side information is included. Four schemes are shown in the figure, using different combinations of reference frames:

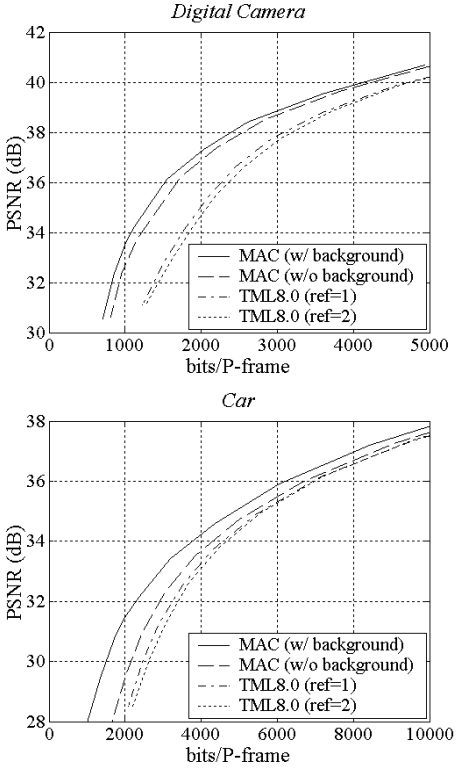

Figure 4. Rate-PSNR curves for model-aided coding (MAC) and TML-8.0. (Top) *Digital Camera* (Bottom) *Car*

A. MAC (modified TML-8.0) using one previous frame and one model frame, with synthesized object and background in the model frame
B. MAC (modified TML-8.0) using one previous frame and one model frame, with only synthesized object in the model frame. No explicit model is used for the background.
C. TML-8.0 using one previous frame.
D. TML-8.0 using two previous frames.

In scheme A, B and D, two reference frames are used and overheads indicating which reference frame is selected for each macroblock need to be encoded. In scheme C, the previous frame is always used hence such overhead is not needed. Scheme D is inferior to scheme C at low bit-rate since one previous frame is sufficient for predic-

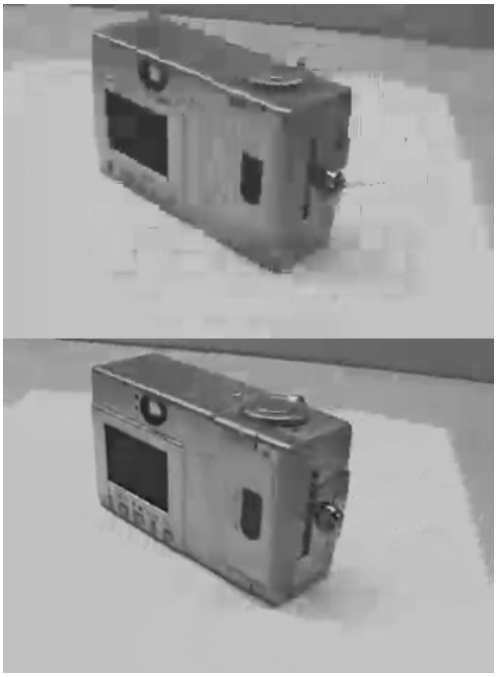

Figure 5. Reconstructed frame 110 of *Digital Camera* using (Top) TML-8.0 (31.1 dB) and (Bottom) MAC (34.2 dB) at 1.2 kbits/P-frame.

Figure 6. Reconstructed frame 10 of Car using (Top) TML-8.0 (28.5 dB) and (Bottom) MAC (31.5 dB) at 2.0 kbits/P-frame.

tion at low quality without the overhead cost of using two previous frames. By comparing schemes B and C, coding efficiency is improved by using the model frame for the foreground object. The improvement in *Digital Camera* is larger than in *Car* due to the varying reflection on the car that cannot be synthesized from the previous frame as well as the fact that the shape model is more accurate in *Digital Camera*. The gap between scheme A and B shows the improvement by the warped background in the model frame. More improvement is observed in *Car* since the background in *Digital Camera* is already simple to encode.

To summarize, up to 30%~40% bit-rate savings are observed with MAC comparing to TML-8.0 at the same quality. This corresponds to an improvement of up to 2-3 dB in PSNR at same bit-rate as shown in Figure 5 and 6. Most of the savings are at lower bit-rates.

From the experimental results, it is observed that the ratio of macroblocks using *COPY* mode is significantly increased in MAC. For MAC, most of the bit-rate saving is due to the lower bit-rate required for motion information, while there is no significant bit-rate reduction for residual coding. The TML-8.0 coder uses sophisticated block-based motion compensation techniques that provides very good prediction and thus reduces the amount of residual coding, but a large portion of bit-rate in TML-8.0 is dedicated to motion vectors. In MAC, those motion vectors can be efficiently represented by 3-D motions and projective transformations resulting in a significant reduction of the total bit-rate.

4 Conclusion

We have presented a model-aided scheme that utilizes 3-D information for video coding. 3-D shape and motion of rigid bodies in video sequences are estimated and efficiently incorporated into a traditional waveform coder. Experiments have shown that bit-rate savings of up to 30%~40% can be achieved compared to TML-8.0 at the same quality.

References

[1] P.Eisert, T. Wiegand, and B. Girod, "Model-aided coding: A new approach to incorporate facial animation into motion-compensated video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 3, pp. 344-358, Apr. 2000.

[2] R. Radke, P. Ramadge, S. Kulkarni, and T. Echigo, "Using view interpolation for low bit-rate video," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 453-456, Oct. 2001.

[3] F. Galpin and L. Morin, "Computed 3D models for very low bitrate video coding," *Proc. SPIE Conf. Visual Communications and Image Processing*, vol. 4310, pp. 255-262, 2001.

[4] G. Calvagno, R. Rinaldo, L. Sbaiz, "Three-dimensional motion estimation of objects for video coding," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 1, pp. 86-97, Jan. 1998.

[5] ITU-T Video Coding Experts Group, H.26L Test Model Long Term Number 8, July 2001, Download via anonymous FTP to: ftp://standard.pictel.com/ftp/video-site/h26L/older_tml/tml8.doc

[6] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 70-84, Feb. 1999.

[7] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images – a review," *Proceedings of the IEEE*, vol. 76, no. 8, August 1988.

[8] *PhotoModeler Pro* demo version via http://www.photomodeler.com

[9] P. Eisert, E. Steinbach, and B. Girod, "Automatic reconstruction of 3-D stationary objects from multiple uncalibrated camera views," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 2, pp. 261-277, March 2000.

[10] P. M. Q. Aguiar and J. M. F. Moura, "Three-dimensional modeling from two-dimensional video", *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1541-1551, Oct. 2001.